REMARKS

Applicant thanks Examiner for the detailed review of the application.

Claim Objections

The Office Action States

2. Claim 4 is objected to for referring to a "second" operating system generated thread and second process, when a "first" operating system generated thread and a "first" process have not yet been introduced to the claims (they are mentioned in Claim 3, but Claim 4 is not dependant upon Claim 3).

Claim 4 has been amended to depend from claim 3, which provides adequate antecedent basis for both "the first thread" and "the first process" of claim 4.

The claim status identifiers of claim 66 and 67 have been updated to Previously Added and Previously Amended, accordingly. Applicant thanks examiner for the interpretation of claim 67, as the current text was newly added and should have been underlined in the previous response. Accordingly, the current full text of claim 67 is included as previously amended without underline.

In addition, applicant has amended claim 70 to remove the second period.

Claim Rejections -35 USC § 102(b)

The Office Action States:

7. Claims 1-2, 5-12, 15-17, 20-28 and 32 are rejected under 35 U.S.C. 102(b) as being anticipated by Lawlor et al. (USPN 5,485,626, herein Lawlor).

"[F]or anticipation under 35 U.S.C. 102, the reference must teach *every aspect* of the claimed invention …" MPEP 706.02 (emphasis added). "The identical invention must be shown *in as complete detail as contained in the … claim*." *Richardson v., Suzuki Motor Co*., 868 F. 2d 1226,

1236, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989) (emphasis added).

Applicant's amended claim 1 includes, "creating, responsive to the programming instruction, a first shared resource thread (shred) that is associated with a first private portion of a first application state and shares a second shared portion of the first application state with at least a second shred, wherein the first application state is a private application state of a first thread, wherein creating the first shred is performed without the intervention of an operating system," (Amended claim 1). The Office Action states that a first private portion of a first application state is inherent in the definition of a thread. Applicant agrees in that traditionally, each thread includes some private storage (private application state) to allow an OS to schedule separate applications or processes, which may have different virtual views of memory, on different threads.

However, as can be seen in applicant's claim 1, this inherent private storage for a thread is included as "a private application state of the first thread," and as a result, "the first private portion of the first application state" in applicants claim 1 shares the private application state of a thread, i.e. is a separate private portion which is both not inherent and is also not described in Lawlor. In contrast to the traditional view and Lawlor, applicant's disclosure discusses essentially further subdivides a thread's application state into private portions, i.e. a first private portion that is associated with a first shred, and a shared portion, i.e. a second shared portion, which is associated with a first shred and a second shred. Note although not specifically stated in claim 1, the second shred may also be associated with a private portion of a thread's application state.

In one embodiment, described in Table 4 of applicant's specification, specific registers of a thread's application state are replicated to form per-shred private portions. Here, each thread includes a single application/system state; part of which is shared between shreds and part of which is replicated to be privately held by each shred. Note that Lawlor does not disclose replicating a portion of a thread's state. In other words, Lawlor does not describe subdividing an application

state of a thread to form private portions of the application state, i.e. shreds, which share another portion of the application state of a thread. In contrast, Lawlor only describes execution of multiple threads on multiple processors, not execution of shreds of a thread. The Office Action infers that "thread," as used by Lawlor includes private portions. This inference illustrates that Lawlor not only does not specifically disclose that threads may have private application states, but that Lawlor does not describe or suggest usage of private and/or shared portions of those private application states, as described in applicant's claim 1.

Applicant's claim 17, includes the element, "a shared register to be **addressable by a user-level instruction**, the shared register to be **directly accessible by at least the first shred and the second shred** to provide communication between the first shred and the second shred," (emphasis added). First, Lawlor specifically describes in many places that "a user cannot access it (object storage) directly at any time," (col. 7 lines16-20), a user (i.e. application program) does not have access to the object repository." (col. 11 lines 13-15), and "Objects do not "reside" in regularly addressable memory and cannot be referenced using regular instructions." Specifically, Lawlor discloses a Send/Receive Queue (SRQ), which The Office Action cites as analogous to applicant's claim 17 element included above, which may be accessed by a send message and a receive message. However, Lawlor specifically describes SRQ as a message queue, which is an object part of the object repository, as illustrated by Fig. 1 (msg. Qs) and described through out the disclosure. As a result, the send and receive messages must be non-user level messages, such as machine generated or other priority program (i.e. not application or user program) generated messages.

Note that applicant specifically makes the distinction in claim 17 that the shared register is to be addressable **by a user level instruction.** In stark contrast, Lawlor explicitly states that these objects are not accessible to the user. As a result, the communication between Lawlor's threads is perform utilizing some priority hardware or priority software, while not allowing user's to access

the SRQ. As can be seen, in the alternative to a user-level instruction, a non-user level instruction, such as Lawlor's send message and receive messages, may be utilized. Applicant's claim 17 does not necessarily preclude usage of non-user level instructions. However, applicant's claim 17 specifically states the register is to be addressable by a user-level instruction, while Lawlor explicitly states that objects, such as the SRQ, are not to be accessible or addressable by users.

Furthermore, the SRQ is a queue by nature. As a result, data or pointer values is/are taken from a register of one thread, placed on the queue, and then de-allocated to a register of another thread. As a result, the register is not truly shared between threads as **directly accessible**, but rather only a queue mechanism to pass information between registers, i.e. register to register communication and not direct loads, stores, moves, etc. from a register.

Claim Rejections -35 USC § 103(a)

"The examiner bears the initial burden of factually supporting any prima facie conclusion of obviousness." MPEP § 2142. It is well established that *prima facie* obviousness is only established when three basic criteria are met. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or combine reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations. *In re Vaeck*, 947 F.2d 488 (Fed. Cir. 1991) (MPEP 2144). The Office Action has failed to meet one or more of these requirements.

The Office Action further states:

35. Claims 35-44, 50, and 62 are rejected under 35 U.S.C. 103(a) as being unpatentable over Lawlor and Galvin, further in view of Patterson et al. (herein Patterson).

Applicant's amended claim 35 includes, "receiving user-level programming instructions to execute a plurality of user-level threads of execution, wherein each of the plurality of user-level threads of execution are to be associated with a private state of an OS-generate thread and to share a system state with the ~~an~~ OS-generated thread." Similar to the discussion above, Lawlor does not disclose a plurality of private states (associated with user-level threads) with a thread, such as an OS-generated thread, but rather only by inference that threads themselves have a private state from other threads. In other words, The Office Action only infers that private states exist between threads, but not that there may be private states within a single thread, as in applicant's claim 35. Additionally, Galvin and Patterson do not disclose private states within threads, but merely reiterate that a thread may be private from another thread, as well as share some memory space or other

resources with another thread.

The Office Action also states:

64. Claims 51, 55-58, and 63-66 are rejected under 35 U.S.C. 103(a) as being unpatentable over Lawlor, in view of Patterson.

Claim 51 includes, "is also capable of concurrently executing multiple shared resource threads (shreds) **associated with one of the OS-generated threads**, wherein each of the multiple shreds are associated with a private state within the OS-generated thread and a shared state of the OS-generated thread," (Claim 51 as amended). Similar to the discussion above, neither Lawlor or Patterson describe shreds within a thread, i.e. each shred of an OS generated thread being associated with a private state **within the OS-generated thread** and a shared state of the OS generated thread.

Claim 55 includes, "a microprocessor, including a plurality of user-level multithreading registers, wherein the registers are addressable by one or more user-level instructions in each of a plurality of user-level threads and are to support communication among the user-level threads," (claim 55). As described above in regards to claim 17, Lawlor explicitly states as part of the novelty, that objects, such as the SRQ to communicate between threads, are not accessible by a user, such as an application. In fact, only privileged hardware, such as P3E 1003 is able to access and address the objects. In direct contrast, as a potential advantage, claim 55 allows for access by user-level non-privileged instructions. A combination of any reference with Lawlor that describes a user-level instruction to access these registers would be impermissible, as it would teach away an essential element of Lawlory, i.e. that these objects are not user accessible. Notwithstanding, Patterson also does not disclose registers capable of being directly accessed by non-privileged user-level instructions, as in applicant's claim 55.

The Office Action also states:

74. Claims 67-71 are rejected under 35 U.S.C. 103(a) as being unpatentable over Lawlor, further in view of Galvin.

Claim 67 includes "a second group of resources, which is to include a replicate of the first group of resources …wherein the first, second, and third group of resources are private from another privileged-level software entity created thread," (redacted claim 67 as amended). Similar to the discussion above, the Office Action relies on the inherent nature of threads to have a private section. However, in claim 67, not only is the privileged software entity private from another privileged-level entity, but there a further private replicated resources, i.e. the first group and the second group, to hold private information within the already private information of the privileged level thread. In other words, even if Lawlor inherently describes private resources between threads, it does not describe private resources of a privileged thread further having replicated resources to hold private shreds within the privileged thread.

Therefore, applicant respectfully submits that independent claims 1, 17, 35, 51, 55, and 67, as well as their dependent claims, are now in condition for allowance for at least the reasons stated above.

If there are any additional charges, please charge Deposit Account No. 50-0221. If a telephone interview would in any way expedite the prosecution of the present application, the Examiner is invited to contact David P. McAbee at (503) 712-4988.

Respectfully submitted,
Intel Corporation

Dated: February 15, 2008

/David P. McAbee/Reg. No. 58,104
David P. McAbee
Reg. No. 58,104

Intel Corporation
M/S JF3-147
2111 NE 25th Avenue
Hillsboro, OR 97124
Tele – 503-712-4988
Fax – 503-264-1729